Pseudo-Time Method for Optimal Shape Design Using the Euler Equations

Angelo Iollo[1]
Dipartimento di Ingegneria Aeronautica e Spaziale
Politecnico di Torino
10129 Torino, Italy

Geojoe Kuruvila[2]
Advanced Transport Aircraft Development
McDonnell Douglas Aerospace
Long Beach, CA 90810 1870

Shlomo Ta'asan[1]
Department of Mathematics
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

In this paper we exploit a novel idea for the optimization of flows governed by the Euler equations. The algorithm consists of marching on the design hypersurface while improving the distance to the state and costate hypersurfaces. We consider the problem of matching the pressure distribution to a desired one, subject to the Euler equations, both for subsonic and supersonic flows. The rate of convergence to the minimum for the cases considered is 3 to 4 times slower than that of the analysis problem. Results are given for Ringleb flow and a shockless recompression case.

[1]This research was supported in part under NASA Contract No. NAS1-19480, while the author was in residence at the Institute for Computer Applications in Science and Engineering, NASA Langley Research Center.

[2]This research was done while the author was with ViGYAN, Inc. as a contractor to NASA Langley Research Center.

1 Introduction

In recent years there has been a renewed interest in optimal design in fluid mechanics. Faster computers and reliable numerical simulations make feasible some of the aerodynamics optimal design problems which are of engineering interest.

The statement above becomes only partially true when we consider either flows governed by Euler or Navier-Stokes equations, or complicated geometrical configurations with many control parameters. For such cases, shape optimization seems to be still not practical due to extremely time consuming computation. The present work aims at a flexible and feasible approach for such intensive computational problems by applying a novel algorithm proposed by Ta'asan in [19].

The problem of finding a shape that achieves given performance has been attacked by means of inverse problem formulations [12],[2],[21],[5]. These methods have in common the advantage of being solved at the same cost of an analysis problem. They are in general not extendable to three dimensions. Moreover the set of problems that can be solved by means of inverse design is limited.

A more general framework is to consider aerodynamics design problems as optimization problems. From the mathematical viewpoint the problem is to find U such that

$$\begin{cases} U \in \mathcal{U} \\ \mathcal{E}(U) \leq \mathcal{E}(V) \qquad \forall V \in \mathcal{U} \end{cases}$$

where $\mathcal{U}$ is a given set and $\mathcal{E}$ is a real-valued functional defined on $\mathcal{U}$.

Shape design optimization problems are tightly related to control of a system governed by partial differential equations where the controls are on the boundary. Lions set in [13] the mathematical framework for such problems. The theory is concerned mainly with linear systems and is devoted "(i) to obtain necessary (or possibly necessary and sufficient) conditions for U to be an extremum (or minimum), (ii) to study the structure and properties of the equations expressing these conditions, (iii) to obtain constructive algorithms amenable to numerical computations for the approximation of U".

Pironneau ([15],[16]) derived an adjoint method for the minimum-drag problem in Stokes flows and subsequently in flows governed by the incompressible Navier-Stokes equations. Since a Navier-Stokes solver was not available, some solutions were obtained using simpler models; see Glowinski and Pironneau [6].

For the Euler equations Jameson proposed in [10] an adjoint method for wing design which makes use of conformal mapping to control the shape of the wing. Iollo, Salas and Ta'asan [9] studied the case of Euler flows with embedded shocks for a one-dimensional case, and discussed the boundary conditions for the adjoint equations. At the shock location it was shown that further conditions are needed for the adjoint equation to be well posed. Subsequently, Iollo and Salas extended these results to two-dimensional flows, and presented computations with higher-order spatial accuracy [8].

The high computational cost for solving optimization problems governed by fluid dynamics equations comes from several sources. The first is the cost of a single analysis, which for the

Navier-Stokes equations in three dimensions is of the order of a few CRAY hours. Another source is the fact that a repeated solution of the flow equations may be required for gradient methods. An additional significant cost may arise from the calculation of the gradient of the functional.

The use of adjoint methods eliminates the unnecessary cost resulting from computation of the gradient, and is much more efficient compared to other methods including finite-differences and sensitivity analysis. It requires the computation of an extra system of partial differential equations (PDEs), namely the costate equations, but the total cost for gradient calculation is independent of the number of design variables. A comparison study of calculating gradients using adjoint methods and finite-difference methods was done by Beux and Dervieux [3]. They also solved pressure reconstruction problems for compressible internal flows, comparing the performances of several algorithms. Flows with embedded shocks were not considered in this work.

The adjoint method, being an efficient method for calculating the gradient, does not address the computational expense related to the number of gradient iterations required to reach the minimum. In general, the number of iterations required to achieve the minimum grows more than linearly with the number of controls used, making infeasible design problems in three dimensions with many design variables.

Ta'asan proposed in [18] an algorithm to reduce the cost of the optimization to that of a single analysis, namely the one shot method. The idea is to solve the flow equations, the costate equations and the optimality condition at the same time. The main idea in that algorithm was to perform the optimization iteration on coarse grids that are used anyway in the multigrid process. Small numbers of design variables were considered in that case.

Ta'asan, Kuruvila and Salas [20] applied this technique to a potential flow, and extended the method to cases of moderate numbers of design variables. Different design variables are associated with different grids depending on the smoothness of the shape functions associated with them, and are updated on these grids. The performance of this algorithm was practically independent of the number of design variables.

Arian and Ta'asan [1] extended the one shot method to infinite-dimensional design space. The main idea of the method was to construct a relaxation that smoothes the errors in the design variables. Application to control problems and shape design problems have demonstrated solution of the full optimization problem in a cost comparable to that of solving the analysis just a few times, independent of the number of design variables (experiments using up to 128 design variables have been done).

Beux and Dervieux [4] proposed a hierarchical strategy in which the number of control parameters is progressively increased performing a multilevel optimization that seems to render the computational cost independent from the number of control parameters.

The drawbacks of the one shot methods are their programming complexity and the fact that their use is limited to multigrid solvers. This was the motivation for the study of a new type of solution strategy for optimization problems governed by PDEs [19]. The goal was to try to get methods that solve the optimization problem in a cost comparable to that of the analysis. The emphasize was on simplicity and flexibility to work in existing frameworks

which do not necessarily involve multigrid methods.

The main observation is the following. The solution of the optimization problem lies on the intersection of the state, costate and design hypersurfaces (in the state, costate and design spaces). Gradient-based methods (including adjoint formulations) can be viewed as marching along the intersection of the state and costate hypersurfaces. This is an expensive process since each step requires the solution of two PDEs. The idea of the pseudo-time method was to perform the marching on the design hypersurface while improving the distance to the other two. The cost of such an iteration per step is significantly smaller than that of gradient-based methods. Its convergence has been shown by Ta'asan in [19] to be independent of the number of design variables.

In the present paper we apply the pseudo-time method to optimization problem using the Euler equations. Using this method the cost of optimization becomes of the same order as that of analysis. Moreover the algorithm may be implemented with no substantial changes to existing codes. Numerical results indicate that the method converges at a rate which is independent of the number of design variables.

2 Problem statement

The Euler equations are given by

$$\mathbf{U}_t + \mathbf{F}_x + \mathbf{G}_y = \mathbf{0} \tag{1}$$

where

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{pmatrix} \quad \mathbf{F} = \begin{pmatrix} \rho u \\ p + \rho u^2 \\ \rho uv \\ u(\rho e + p) \end{pmatrix} \quad \mathbf{G} = \begin{pmatrix} \rho v \\ \rho uv \\ p + \rho v^2 \\ v(\rho e + p) \end{pmatrix}$$

with

$$\begin{aligned} \rho &= \text{density} \\ u &= \text{x-component of the velocity vector} \\ v &= \text{y-component of the velocity vector} \\ e &= \text{specific total energy} \\ p &= \text{pressure} \\ a &= \text{speed of sound} \\ \gamma &= \text{ratio of specific heats} \\ \kappa &= \frac{\gamma - 1}{2} \end{aligned}$$

and $p = \kappa\rho(2e - u^2 - v^2)$. Furthermore let

$$\frac{\partial \mathbf{F}}{\partial \mathbf{U}} = \mathbf{A}(\mathbf{U}) \quad \text{and} \quad \frac{\partial \mathbf{G}}{\partial \mathbf{U}} = \mathbf{B}(\mathbf{U}) \tag{2}$$

We assume that these equations are defined on a domain Φ which includes a sub-domain Ω whose boundary is denoted by Γ. On the boundary we define a curvilinear coordinate s and a normal $\mathbf{n} = (n_x, n_y)$ pointing outward. A real valued functional $\mathcal{E}(\Gamma, V(\Gamma))$ is given, where $V(\Gamma)$ is the solution of the Euler equations with boundary conditions on Γ. The optimization problem that we study is

minimize the functional $\mathcal{E}(\Gamma, V(\Gamma))$ *over all the admissible shapes of the boundary* Γ.

We focus on the following model problem. The sub-domain Ω is represented by a nozzle; see Fig. 1. At the inlet, total pressure, total temperature and the ratio $\sigma = v/u$ are fixed. At the outlet, if the flow is subsonic, the static pressure is fixed and at the solid walls the impermeability condition $un_x + vn_y = 0$ is enforced. The upper wall is kept fixed. The lower wall Θ is represented by mean of the parameterization

$$y(\Theta) = \sum_i \alpha_i f_i(x) \tag{3}$$

where the functions $f_i(x)$ are some shape functions and $\alpha = (\alpha_1, ..., \alpha_i, ...)$ is the corresponding set of shape coefficients. Given a desirable lower wall pressure distribution $p^*(x)$ and denoting by $p^\Theta(x)$ the actual one on the lower wall, the optimization problem consists in finding a set of shape coefficients α_i such that the functional

$$\mathcal{E} = \frac{1}{2}\int_a^b (p^\Theta - p^*)^2 dx \tag{4}$$

is minimized.

3 Optimality conditions

The optimality conditions are derived by introducing Lagrange multipliers and considering the augmented functional

$$\mathcal{L}(\mathbf{U}, \alpha, \mathbf{\Lambda}, \mu) = \mathcal{E}(\alpha, \mathbf{U}) + \int_\Omega {}^t\mathbf{\Lambda}(\mathbf{A}\mathbf{U}_x + \mathbf{B}\mathbf{U}_y)d\Omega + \int_\Theta \mu\rho\mathbf{V}\cdot\mathbf{n}ds \tag{5}$$

where $\mathbf{V} = (u, v)$. The vector $\mathbf{\Lambda}(x, y) = {}^t(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$, and the scalar $\mu(s)$ are the Lagrange multipliers.

Calculating the variation of the functional $\mathcal{L}$ with respect to the variation of the functions $\mathbf{U}$, $\mathbf{\Lambda}$, μ and the parameters α_i respectively, we obtain (see [8])

$$\begin{aligned}\delta\mathcal{L}_U &= \int_a^b \left.\frac{\partial p}{\partial \mathbf{U}}\right|_\Theta (p^\Theta - p^*)\widetilde{\mathbf{U}}dx + \int_\Gamma {}^t\mathbf{\Lambda}(\mathbf{A}n_x + \mathbf{B}n_y)\widetilde{\mathbf{U}}ds + \\ &- \int_\Omega ({}^t\mathbf{\Lambda}_x\mathbf{A} + {}^t\mathbf{\Lambda}_y\mathbf{B})\widetilde{\mathbf{U}}d\Omega + \int_\Theta \mu\mathbf{n}\frac{\partial\rho\mathbf{V}}{\partial\mathbf{U}}\widetilde{\mathbf{U}}ds\end{aligned} \tag{6}$$

where

$$\frac{\partial p}{\partial \mathbf{U}} = 2k\left(\frac{u^2+v^2}{2}, -u, -v, 1\right) \quad \text{and} \quad \frac{\partial \rho \mathbf{V}}{\partial \mathbf{U}} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

and

$$\delta\mathcal{L}_\Lambda = \int_\Omega {}^t\tilde{\mathbf{\Lambda}}(\mathbf{A}\mathbf{U}_x + \mathbf{B}\mathbf{U}_y)d\Omega \tag{7}$$

$$\delta\mathcal{L}_\mu = \int_\Theta \tilde{\mu}\rho\mathbf{V}\cdot\mathbf{n}ds \tag{8}$$

$$\begin{aligned}\delta\mathcal{L}_\alpha = & \sum_i \left(\int_a^b \left.\frac{dp}{dy}\right|_\Theta (p^\Theta - p^*)\, f_i \; dx + \int_\Theta {}^t\mathbf{\Lambda}(\mathbf{A}\mathbf{U}_x + \mathbf{B}\mathbf{U}_y)\, f_i \, \cos\theta \, ds + \right. \\ & + \int_\Theta \mu \frac{\partial(\rho\mathbf{V})}{\partial y}\cdot\mathbf{n}\, f_i \, ds - \int_\Theta \mu\rho\mathbf{V}\cdot\mathbf{t}\,\frac{df_i}{dx}\,\cos^2\theta\, ds \; + \\ & \left. + \int_\Theta \mu\rho\mathbf{V}\cdot\mathbf{n}\,\frac{df_i}{dx}\,\sin\theta\cos\theta\, ds \right)\tilde{\alpha}_i \end{aligned} \tag{9}$$

where θ is the angle between the normal $\mathbf{n}$ and the y-axis, $\mathbf{t} = (-n_y, n_x)$, and $\widetilde{\mathbf{U}}$, $\tilde{\mathbf{\Lambda}}$, $\tilde{\mu}$ and $\tilde{\alpha}_i$ are the variations of the corresponding arguments.

At the minimum of the functional, for all the possible choices of the functions $\widetilde{\mathbf{U}}$, $\tilde{\mathbf{\Lambda}}$, $\tilde{\mu}$ and of the parameters $\tilde{\alpha}$, we must have

$$\delta\mathcal{L}_U = \delta\mathcal{L}_\Lambda = \delta\mathcal{L}_\mu = \delta\mathcal{L}_\alpha = 0. \tag{10}$$

Therefore, we have

$$\delta\mathcal{L}_\Lambda = 0 \;\Leftrightarrow\; \mathbf{A}\mathbf{U}_x + \mathbf{B}\mathbf{U}_y = 0 \text{ on } \Omega$$

and

$$\delta\mathcal{L}_\mu = 0 \;\Leftrightarrow\; \rho\mathbf{V}\cdot\mathbf{n} = 0 \text{ on } \Theta$$

which are the Euler equations and boundary conditions. Furthermore

$$\delta\mathcal{L}_U = 0 \;\Leftrightarrow {}^t\mathbf{A}\mathbf{\Lambda}_x + {}^t\mathbf{B}\mathbf{\Lambda}_y = \mathbf{0} \text{ on } \Omega \tag{11}$$

and

$$\left.\frac{\partial p}{\partial \mathbf{U}}\right|_\Theta (p^w - p^*)\,\cos\theta \; + {}^t\mathbf{\Lambda}(\mathbf{A}n_x + \mathbf{B}n_y) + \mu\mathbf{n}\,\frac{\partial\rho\mathbf{V}}{\partial\mathbf{U}} = \mathbf{0} \text{ on } \Theta \tag{12}$$

where

$$\mu = -\left[\lambda_1 + u\lambda_2 + v\lambda_3 + (\gamma e - kV^2)\lambda_4\right] \tag{13}$$

For the boundary condition on inlet and outlet we refer the reader to [8]. Given $\mathbf{U}$, the set of costate eqs.(11-13) determine uniquely $\mathbf{\Lambda}$ in Ω and μ on Θ. Finally, given α and knowing $\mathbf{U}$ and $\mathbf{\Lambda}$, we can calculate from eq.(9)

$$\frac{\partial\mathcal{L}}{\partial\alpha_i} = \int_a^b \left.\frac{dp}{dy}\right|_\Theta (p^\Theta - p^*)\, f_i \, dx + \int_\Theta {}^t\mathbf{\Lambda}(\mathbf{A}\mathbf{U}_x + \mathbf{B}\mathbf{U}_y)\, f_i \, \cos\theta \, ds +$$

$$+ \int_\Theta \mu \frac{\partial(\rho \mathbf{V})}{\partial y} \cdot \mathbf{n}\, f_i\, ds - \int_\Theta \mu \rho \mathbf{V} \cdot \mathbf{t}\, \frac{df_i}{dx}\, \cos^2\theta\, ds +$$
$$+ \int_\Theta \mu \rho \mathbf{V} \cdot \mathbf{n}\, \frac{df_i}{dx}\, \sin\theta \cos\theta\, ds \qquad (14)$$

In case of a shock occuring in the flow field, we split the domain of integration by means of a curve Υ that coincides with the shock where it exists. Then we follow the same derivation presented so far on each of the two sub-domains, regarding Υ as a boundary; see [8]. The resulting extra condition for $\mathbf{\Lambda}$ on the shock is $\mathbf{\Lambda} = \mathbf{0}$. It should be noted that if the shocks are not treated properly, the problem of solving the costate equations with boundary conditions is not well-posed. Jameson presented in [11] results for transonic flows over airfoils where the wave-drag is minimized. He does not use any special treatment for the shock but the costate equations converge. This is due to the fact that the scheme that he uses for solving the Euler equations smears the shocks over several grid points, due to artificial viscosity.

4 Pseudo-time optimization method

There are many methods for obtaining the minimum of the functional $\mathcal{L}$ knowing its gradient with respect to the controls. Adjoint methods involve the following steps:

1. Start with a set α_i of shape coefficients
2. Enforce $\delta\mathcal{L}_\Lambda = 0$ and $\delta\mathcal{L}_\mu = 0$ by finding a $\mathbf{U}$ that satisfies the steady state Euler equations and boundary conditions
3. Enforce $\delta\mathcal{L}_U = 0$ by finding a $\mathbf{\Lambda}$ that satisfies the costate equations and boundary conditions
4. Calculate $\nabla_\alpha \mathcal{L}$, if it is 0 we have found the minimum, otherwise
5. Update α with $\nabla_\alpha \mathcal{L}$, using a proper stepsize.
6. Restart from 2 until $\nabla_\alpha \mathcal{L} = 0$.

The need to repeat steps 2 and 3 above many times can become prohibitively expensive for geometrically complex configurations requiring computational power near the limits of present capabilities.

Ta'asan proposed in [19] an efficient way of solving the optimization problem. The main observation is the following. The solution of the optimization problem lies on the intersection of the state, costate and design hypersurfeces (in the state, costate and design spaces). Gradient based methods (including adjoint formulations) can be viewed as marching along the intersection of the state and costate hypersurfaces. This is an expensive process since each step requires the solution of two PDEs. The idea of the pseudo-time method was to perform the marching on the design space while improving the distance to the other two. Compare Fig. 2 and 3. The cost of such an iteration per step is significantly smaller than

that of gradient based methods. Its convergence has been shown by Ta'asan in [19] to be independent of the number of design variables.

The design equation for a wide class of problems, including the one considered here, is defined on the boundary only. Thus, it can be viewed as an extra boundary condition, and the design variables as the additional variables to solve for. In some cases the design equation can be solved for the design variables and a simple implementation of the above idea exists. In other cases the design equation, viewed as an equation for the design variables keeping the state and costate fixed, may be singular and a more involved implementation is required. This is the case for the problem considered here. In such cases it is necessary to solve for the design variables together with the state and costate variables in a small vicinity of the boundary $\mathcal{S}$.

Thus, at each step of computation on the entire field, the design equation is satisfied together with the boundary conditions for the state and costate equations. The solution on the entire field Ω affects the result of the optimization on $\mathcal{S}$ through the values of $\mathbf{U}$ and $\mathbf{\Lambda}$ on the auxiliary boundary Ψ; see Fig. 4.

The algorithm is as follows:

1. Start with a tentative set of α_i.
2. March in time, on the entire field, the state equation a few steps.
3. March in time, on the entire field, the costate equation a few steps.
4. Solve in $\mathcal{S}$ the state equation with its boundary conditions, the costate equation with its boundary conditions and compute $\nabla_\alpha \mathcal{L}$.
5. If $\nabla_\alpha \mathcal{L} = 0$, restart from step 2, repeating steps 3 and 4 until the state and costate equations are converged on the entire field. Otherwise take $\alpha_i^{n+1} = \alpha_i^n + f(\nabla_\alpha \mathcal{L})$ and go to 4.

We took $\alpha_i^{n+1} = \alpha_i^n - a\,\nabla_\alpha \mathcal{L}$, where a is a parameter. One could try to solve the problem on the boundary, in step 4 above, using a direct solver. The way we propose here has the advantage of being a simple modification of adjoint method, and therefore can be easily implemented.

5 First optimization experiments

We introduce a discrete grid defined as $(x_l, y_m) = (x_o + l\Delta x, y(\Theta) + m\Delta y)$ where Δx is constant and Δy is a constant fraction of the local height of the nozzle; see Fig.5.

The steady solution of the Euler equations is obtained with a *time-dependent* technique, in the frame of an explicit finite volume code. The conservative variables $\mathbf{U}$ are computed at the cell centers, and the fluxes $\mathbf{F}$ and $\mathbf{G}$ are evaluated at the cell interfaces using the approximate Riemann solver in [14]. Higher-order accuracy is achieved using an Essentially

Non-Oscillatory scheme [7]. The flow field values at cell interfaces, used as initial conditions for the Riemann problem, are reconstructed by means of a linear interpolation and using a minimod limiter. The amplitude of the integration step is chosen according to the CFL condition.

The costate equations are discretized on the same grid presented above. Since they have no conservative form, the numerical solution is obtained using the finite-difference scheme proposed in [8].

The computations are performed on a 40×20 grid. Total pressure and total temperature at the inlet are taken to be unity and $\sigma(0, y) = 0$. At the outlet the static pressure depends on the test-case considered. For the lower wall ordinate $y(\Theta)$ we have

$$y(\Theta) = \begin{cases} 0 & \text{if } -0.5 \leq x < 0 \\ \sum_{i=1}^{4} \alpha_i \, x^{i+1} \, (x-1)^2 & \text{if } 0 \leq x < 1 \\ 0 & \text{if } 1 \leq x < 1.5 \end{cases}$$

We try to recover the pressure distribution obtained with the Euler solver, corresponding to the set of shape coefficients $\alpha = (0, 2, 2, 0)$. This means that the functional is 0 at the minimum. The outlet pressure is such that the flow presents a relevant shock at re-compression.

Figure 6 shows the functional values at each step of computation. Figure 7 shows the the convergence history of the state equation, computed to second-order accuracy, and the convergence history of the costate equation. Finally in Fig. 8, we present the starting pressure distribution and the one obtained at the end of the optimization procedure.

The practicability of this approach depends on the rate of convergence to the minimum. In fact the state and costate equations converge to the steady solution with a less favorable rate compared to that of a simple analysis. It is easily seen that since the shape is changing, the flow field must change accordingly and so must the residuals. Figure 9 shows a comparison of the residuals for the state equations in the case of a simple analysis to the residuals in the optimization case. The convergence rate is 3 to 4 times slower in the optimization case. Considering the costate equations, the cost of the optimization procedure turns to be of the order of 10 analyses; using the first of the two algorithms presented in Section 4 the factor of proportion is 100 to 200 depending on the updating strategy used. The CPU time needed on a DEC 3000/500 is 18 minutes with the algorithm presented. For the first algorithm of Section 4, 6 hours of CPU time were needed.

The present rate of convergence could be improved by changing the way of updating the grid. In fact, close to the minimum, the entire grid is perturbed to update only the boundary. We believe that, close to the minimum, the rate of convergence can be improved by updating only the boundary points of the grid. In fact, the small difference between the desired pressure and the one obtained is due to the fact that close to the minimum the convergence rate of the equations is reduced. Therefore the pressure p^* is obtained asymptotically.

6 Optimal shape for compressible flows

The following examples represent situations for which the optimal solution is not generated with the same algorithm we used to study the optimization problem. In the first case, we recover a pressure distribution known theoretically and compare the shape obtained with the theoretical one.

The Ringleb flow (see [17]) is a two-dimensional steady compressible isentropic flow, where subsonic, transonic and supersonic regions are represented. It describes a 180^o-turn of a compressible flow; all the exact values of the flow properties are given by simple formulas dependent on the stream function and on the Mach number. We consider the portion of the flow confined between two streamlines, which may be regarded as solid walls. The maximum Mach number on the bottom streamline is 1.6 and the minimum 0.8. On the top streamline the maximum Mach number is 0.8. The theoretical Mach number isocontours for such a flow are shown in Fig. 10.

The Ringleb pressure distribution on the bottom wall is taken as the desired distribution that we want to achieve.

The lower wall is described by the following parameterization:

$$y(\Theta) = r_0 + (r_1 - r_0) \sum_{i=1}^{4} \alpha_i \sin(i\,\pi\,\frac{\theta - \theta_0}{\theta_1 - \theta_0})$$

where r_0 is the distance, measured from the point of intersection of the lines from the inlet and the outlet, to the first point on the lower wall and r_1 is the distance to the last point. The angles θ_0 and θ_1 are relative to the first and last point respectively, and are measured from the line from the inlet.

In principle, if we try to recover the pressure distribution on the lower wall, the solution is out of the design space. We don't have any *a priori* knowledge of the values that the α_i will assume and how close to the desired pressure we can get.

In Fig. 11 it is seen that no visible difference can be appreciated between the theoretical wall shape and the optimal shape found. The points representing the two solutions do not overlap since they are computed on two different grids. In this case the functional eq.(4) is $6.70 \cdot 10^{-5}$ after 500 iterations of steps 2. and 3. of the second algorithm proposed. See Fig. 12. The CPU time required for this case is about 20 minutes.

In the second case considered, we are concerned with a convergent nozzle. The lower wall is represented by a parameterization similar to that above. The inlet Mach number is 2.2 and the grid is 80×40. The starting configuration with pressure contourlines is shown in Fig. 13.

A relevant shock is present in the flow field and our objective is to eliminate it by requiring a smooth compression at the lower wall. The smooth pressure distribution is not perfectly attained, as is seen in Fig. 14. Nevertheless the recompression appears to be smooth and the shock is eliminated from the flow field (Fig. 15). These results are obtained after 200 iterations in about 35 minutes of CPU. The functional is decreased from $3.75 \cdot 10^{-3}$ to

$1.35 \cdot 10^{-4}$. The computation has been pursued for 2000 iterations and the functional value remained unchanged.

Finally, an experiment using 8 shape coefficients is performed. In Fig. 16 it is seen that the convergence rates of the state and costate equations are not affected. The functional minimum is therefore attained with the same number of iterations as in the case of 4 shape coefficients.

7 Conclusions

The pseudo-time method was applied to optimization problems governed by the Euler equations in two dimensions. The problem of matching the pressure distribution to a desired one was considered, both for subsonic and supersonic flows. The rate of convergence to the minimum for the cases considered is 3 to 4 times slower compared to that of the analysis problem. Results were obtained for Ringleb flow and a shockless recompression case. The algorithm could be implemented with no substantial changes to existing adjoint based codes. Numerical results indicate that the method converges at a rate which is independent of the number of design variables. The method offers a powerful and inexpensive tool for the study of non-intuitive configurations for aerodynamic design.

Aknowledgments

Many discussions with Manuel D. Salas of NASA Langley and with Professor Maurizio Pandolfi of the Politecnico di Torino contributed to the formulation of the algorithm proposed.

References

[1] E. Arian and S. Ta'asan. Multigrid one shot methods for optimal control problems: infinite dimensional control. Technical Report 94-52, ICASE, 1994.

[2] F. Bauer, P. Garabedian, and D. Korn. *Supercritical Wing Sections*. Springer-Verlag, Berlin, 1972.

[3] F. Beux and A. Dervieux. Exact-gradient shape optimization of a 2d Euler flow. *Finite Elements in Analysis and Design*, vol. 12:281–302, 1992.

[4] F. Beux and A. Dervieux. A hierarchical approach for shape optimization. Technical Report 1868, INRIA, 1993.

[5] P. Chaviaropoulos, V. Dedoussis, and K.D. Papailiou. Single-pass method for the solution of the inverse potential and rotational problems part ii: fully 3-d potential theory and applications. Technical Report 803, AGARD, 1994.

[6] R. Glowinski and O. Pironneau. On the numerical computation of the minimum-drag profile in laminar flow. *Journal of Fluid Mech.*, vol. 72, 1975.

[7] A. Harten, B. Engquist, and S.R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, iii. *Journal of Computational Physics*, vol. 71, 1987.

[8] A. Iollo and M.D. Salas. Contribution to the optimal shape design of two-dimensional internal flows with embedded shocks. Technical Report 95-20, ICASE, 1995.

[9] A. Iollo, M.D. Salas, and S. Ta'asan. Shape optimization governed by the Euler equations using an adjoint method. Technical Report 93-78, ICASE, 1993.

[10] A. Jameson. Aerodynamic design via control theory. Technical Report 88-64, ICASE, 1988.

[11] A. Jameson. Optimum aerodynamic design via boundary control. Technical Report 803, AGARD, 1994.

[12] M.J. Lighthill. A new method of two dimensional aerodynamic design. *ARC Rand M 2112*, 1945.

[13] J.L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*. Springer-Verlag, Berlin, 1971.

[14] M. Pandolfi. A contribution to the numerical prediction of unsteady flows. *AIAA J.*, vol. 22(5), 1984.

[15] O. Pironneau. On optimum profiles in stokes flow. *Journal of Fluid Mech.*, vol. 59, 1973.

[16] O. Pironneau. On optimum design in fluid mechanics. *Journal of Fluid Mech.*, vol. 64, 1974.

[17] F. Ringleb. Exakte Loesungen der Differentialgleichungen einer adiabatischen Gasstroemung. *ZAMM*, vol. 20, 1940.

[18] S. Ta'asan. One shot methods for optimal control of distributed parameters systems. Technical Report 91-2, ICASE, 1991.

[19] S. Ta'asan. Pseudo-time methods for constrained optimization problems governed by PDE. Technical Report 95-32, ICASE, 1995.

[20] S. Ta'asan, G. Kuruvila, and M.D. Salas. Aerodynamic design and optimization in one shot. In *30th Aerospace Sciences Meeting and Exhibit, AIAA 92-005*, Jan. 1992.

[21] L. Zannetti. A natural formulation for the solution of two-dimensional or axisymmetric inverse problems. *Int. J. Numerical Methods in Engineering*, vol. 22:451–463, 1986.

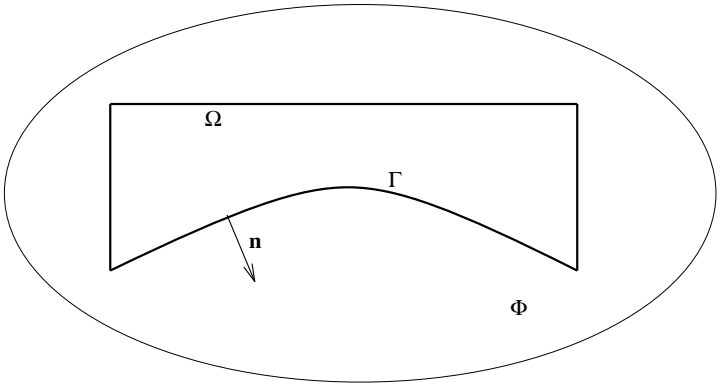

Figure 1: Model Problem.

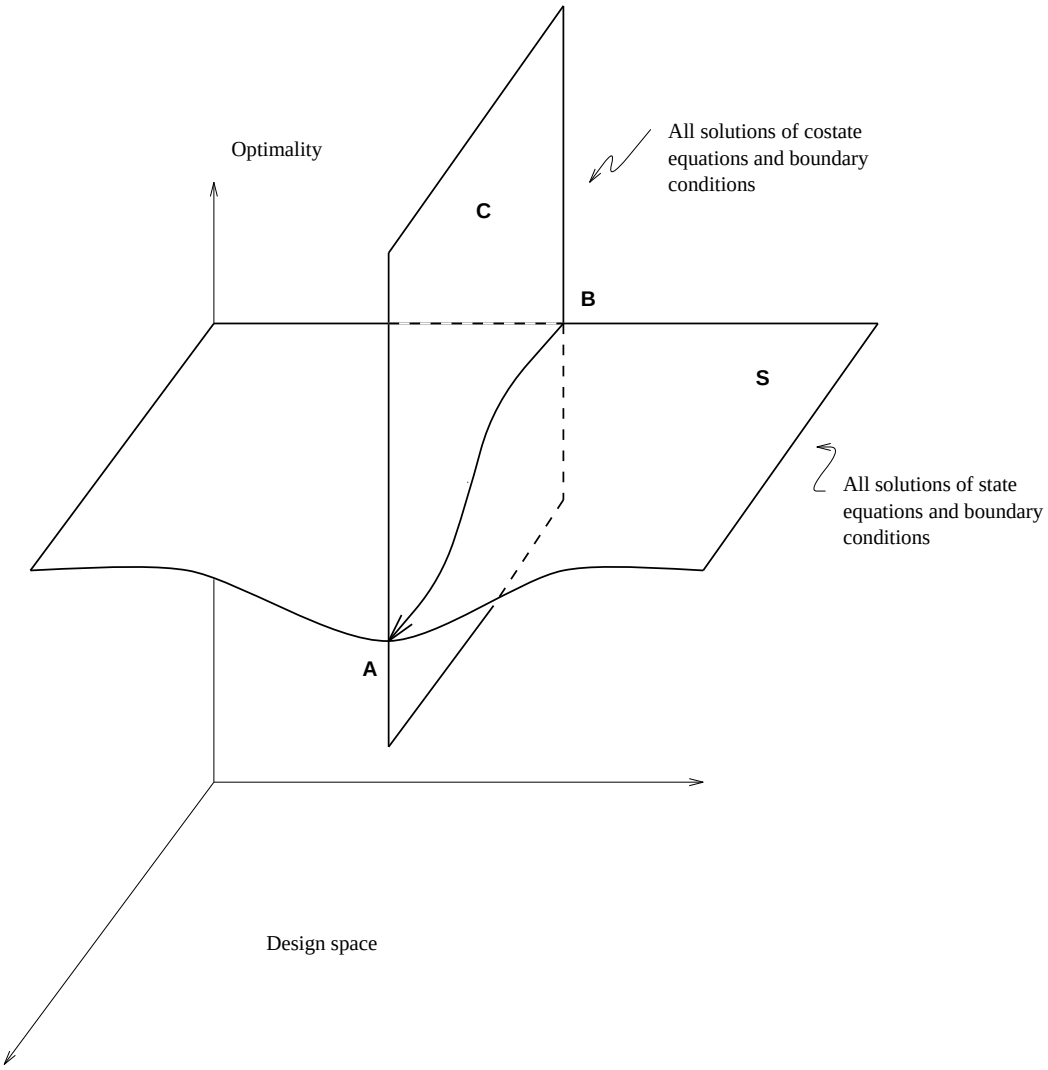

Figure 2: Point A represents the desired optimum, point B the starting configuration. In the standard adjoint method, point A is reached by following a narrow path corresponding to the intersection of plane S and C. At each step along $A - B$, the state and costate equations are iterated many times.

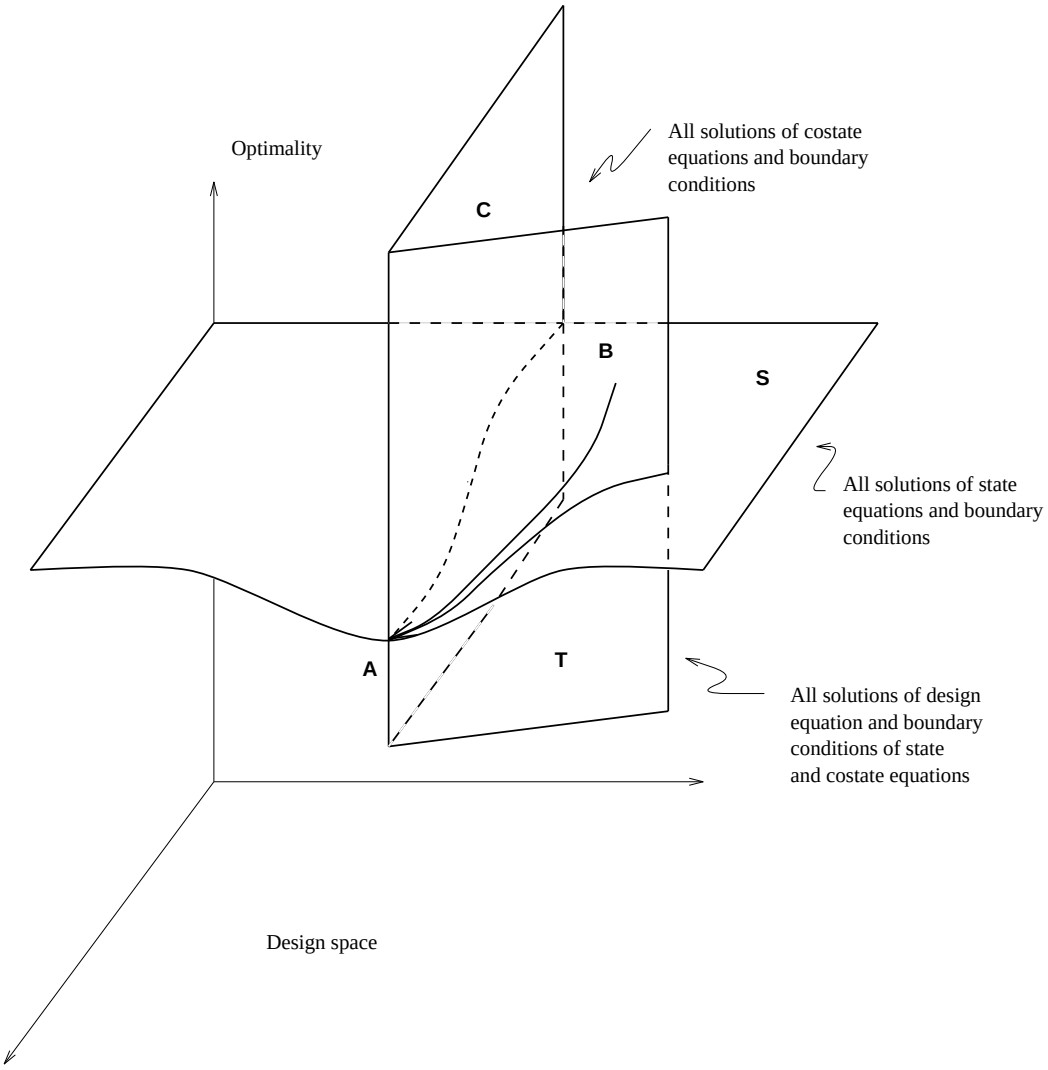

Figure 3: In the new approach a new path, $A-B$, is taken lying on the plane T representing all solutions to the design equation and the boundary conditions of state and costate equations. The computational cost of working on this plane is equivalent to solving a problem one space dimension less than the original problem. The solution of the state and costate equations is achieved only when point A is reached.

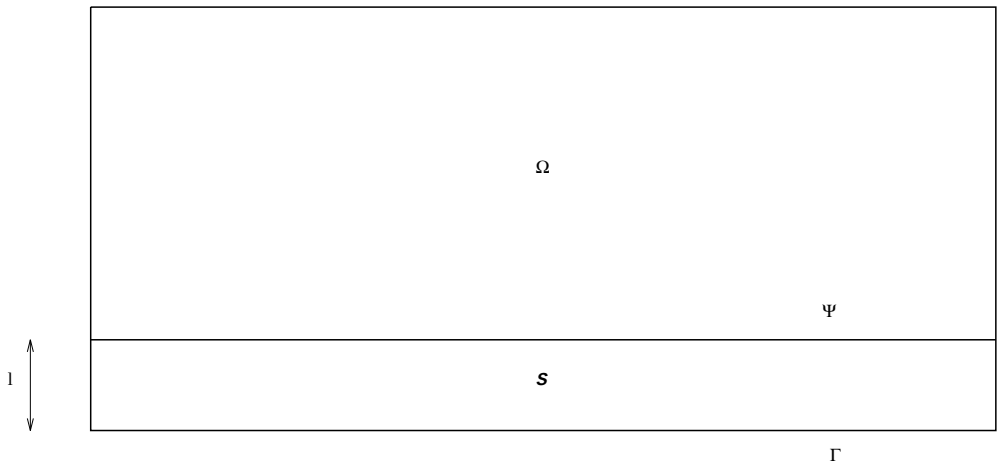

Figure 4: Domain of integration and auxiliary boundary Ψ.

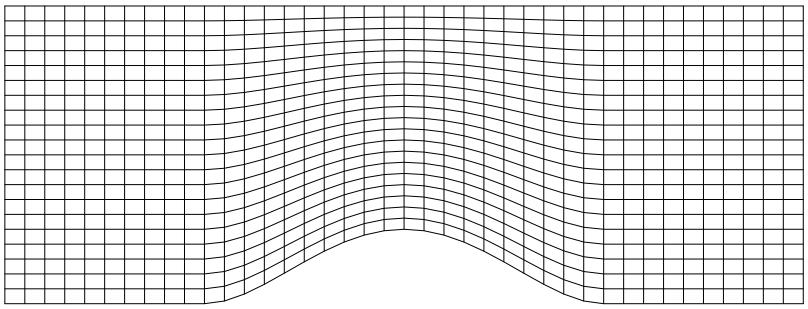

Figure 5: Discrete grid.

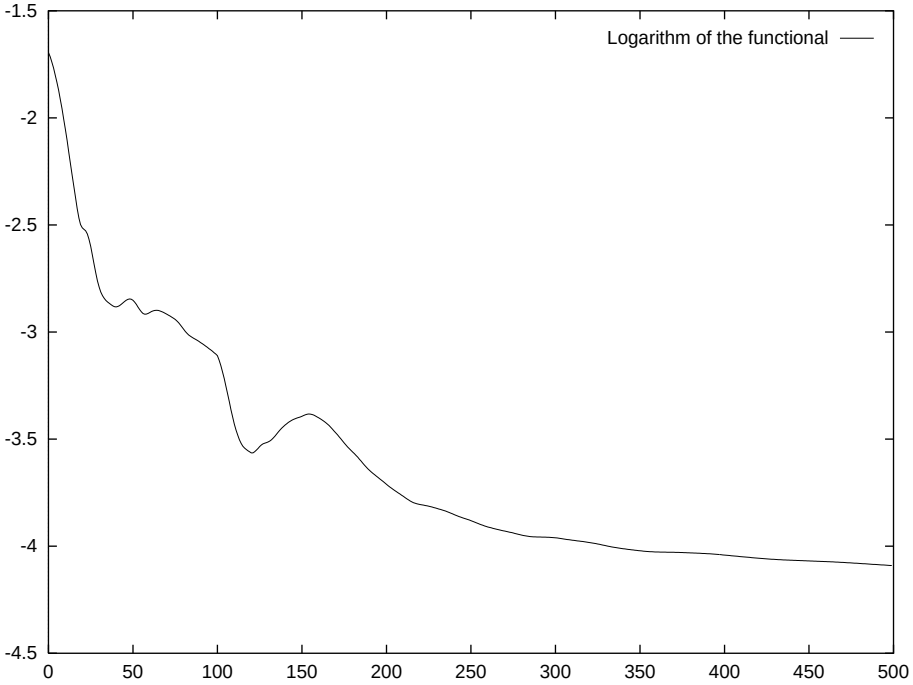

Figure 6: Logarithm of the functional versus the number of iterations on the entire field.

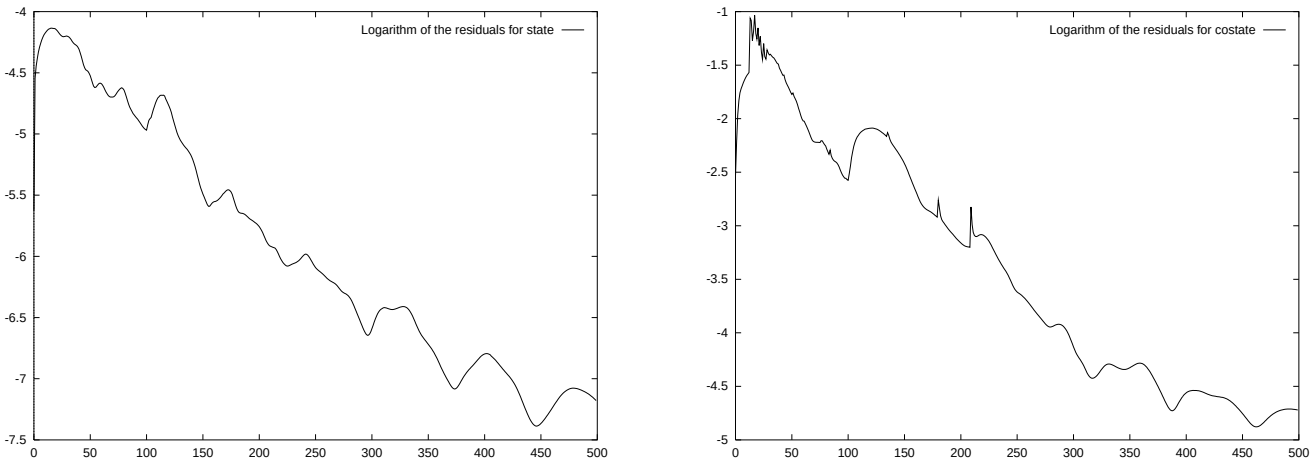

Figure 7: Convergence history for state and costate equations. Logarithm of the residuals.

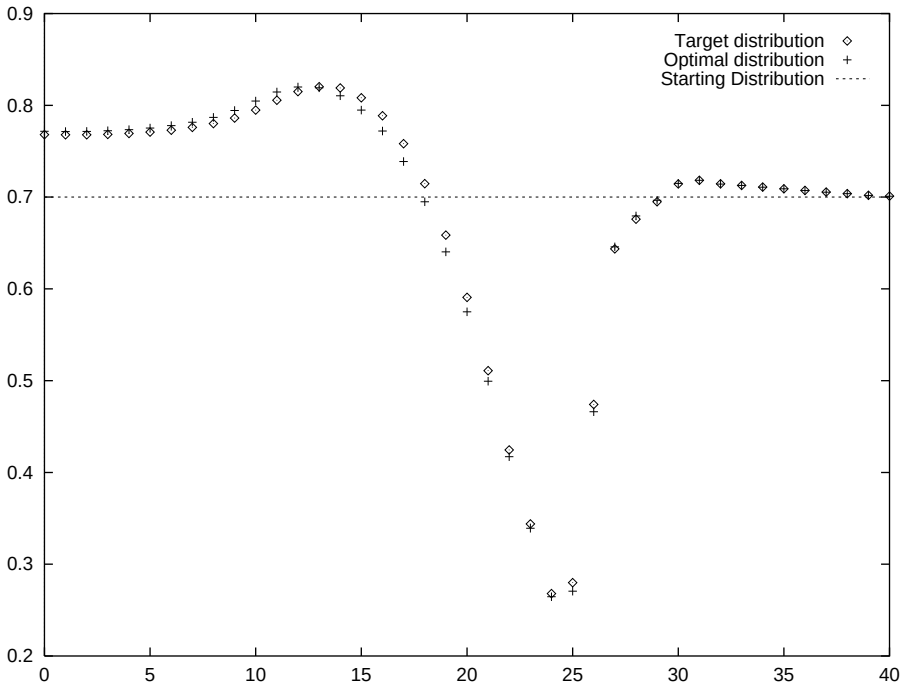

Figure 8: Target pressure distribution and optimal one. Starting pressure: constant distribution at 0.7 reference value.

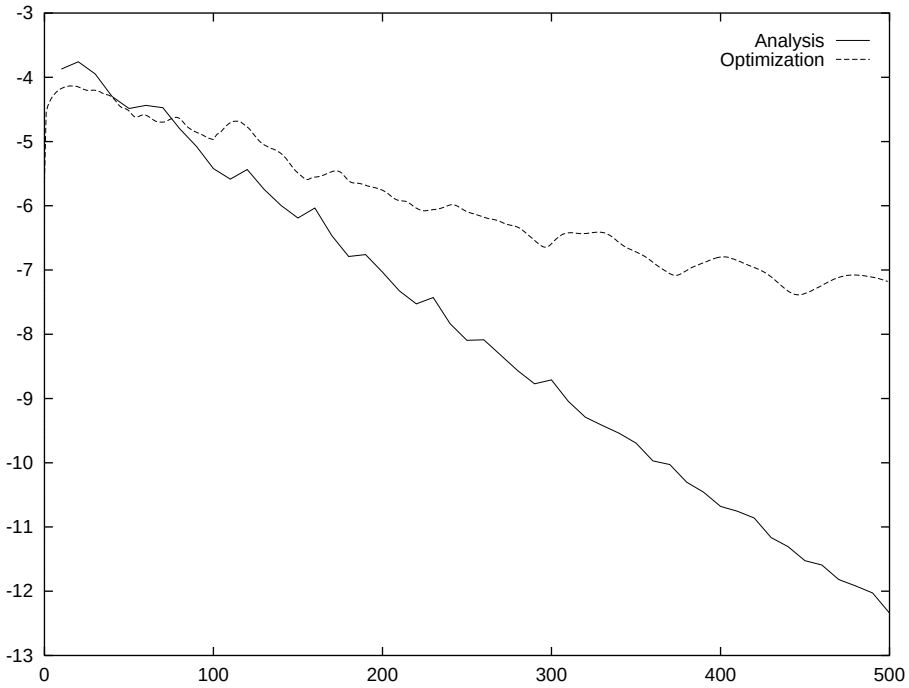

Figure 9: Convergence history of state equations in a simple analysis compared to the convergence of the state equations in an optimization procedure.

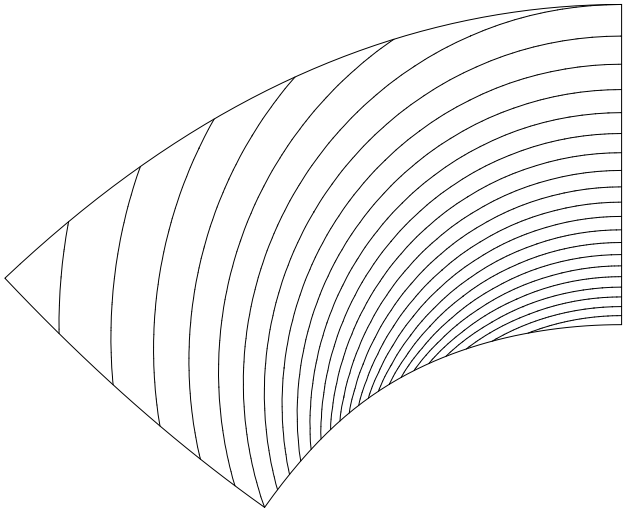

Figure 10: Ringleb flow: Mach number isocontours.

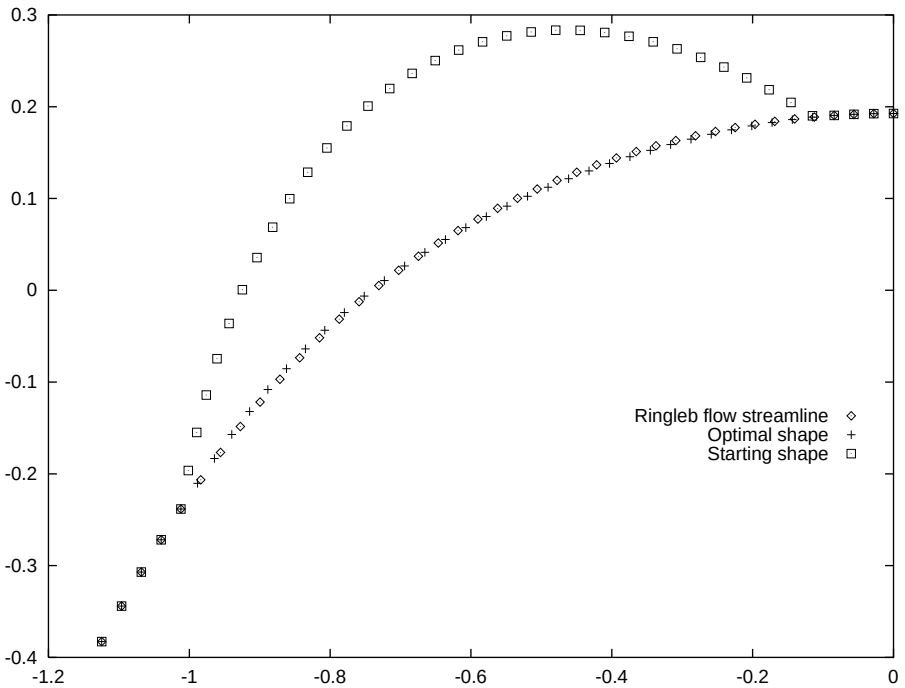

Figure 11: Ringleb flow: starting configuration, theoretical solution and optimal shape.

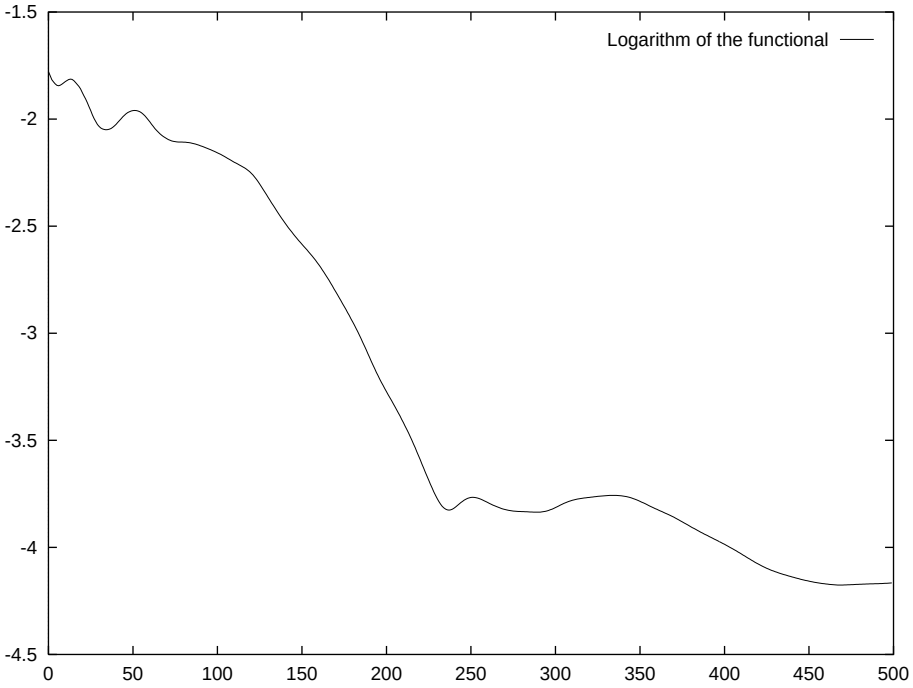

Figure 12: Logarithm of the functional versus the number of iterations on the entire field.

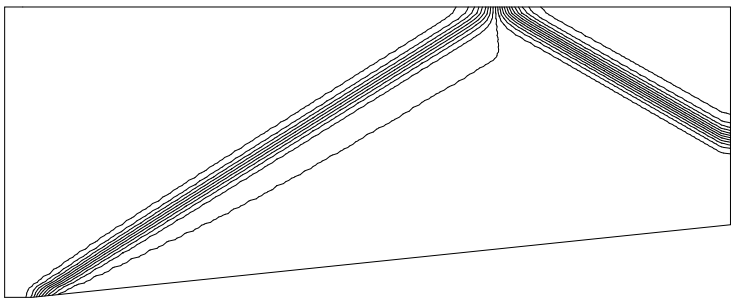

Figure 13: Convergent nozzle starting configuration: pressure isocontours.

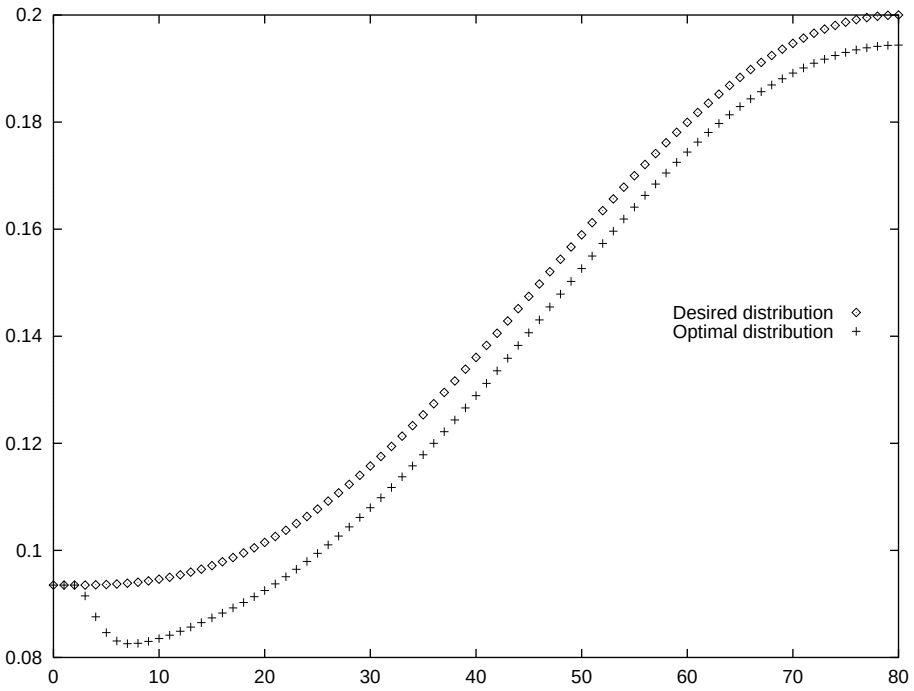

Figure 14: Convergent nozzle: desired pressure distribution and optimal one.

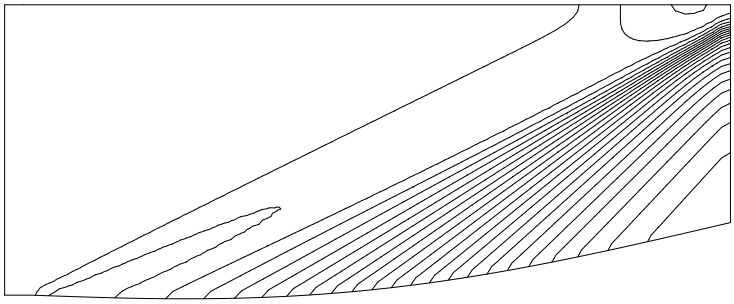

Figure 15: Convergent nozzle optimal solution: pressure isocontours.

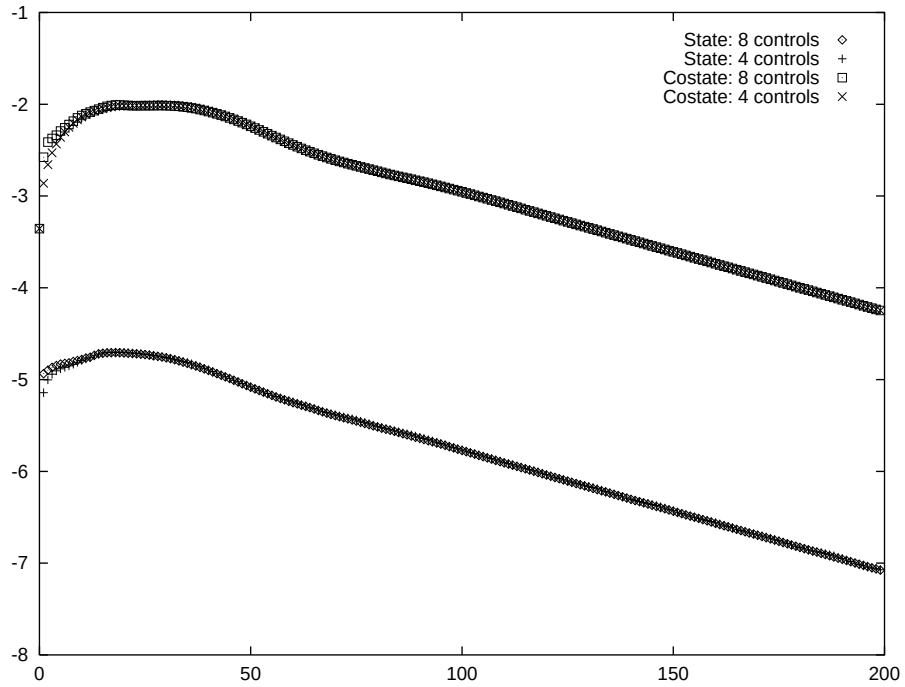

Figure 16: Convergent nozzle. State and costate convergence history: 4 shape coefficients versus 8 shape coefficients.